

VIDEO OUTPUT CONTROLLER AND VIDEO CARD

CROSS REFERENCE TO RELATED APPLICATIONS

This application is based upon and claims the
5 benefit of priority from the prior Japanese Patent
Application No. P2003-328272 filed on September 19,
2003; the entire contents of which are incorporated
herein by reference.

10 BACKGROUND OF THE INVENTION

1 . Field of the Invention

The present invention relates to a video output
controller and a video card, and particularly, to a video
output controller capable of programming a graphics
15 processor.

2 . Description of Related Art

Japanese Patent Laid Open Publication (Kokai) No.
2001-84217 discloses a DMA (direct memory access)
20 circuit having a memory to store a plurality of source
addresses and a source address register to hold a first
one of the source addresses stored in the memory.

Japanese Patent Laid Open Publication (Kokai) No.
2002-132706 discloses a DMA transfer device having a
25 transfer address storing unit to store a first address of a
data block containing data for a plurality of lines to be

DMA-transferred in response to an instruction from a CPU and a storage unit to store the number of data lines to be transferred.

Specification of United States Patent No.
5 6202106B1 discloses an intelligent DMA controller (IDMAC) having a parameter block structure that includes an initial memory location and at least one pointer, which is based on the initial memory location, for a plurality of parameter locations (such as source
10 addresses and destinations).

These DMA devices are constituted by hardware, and therefore, only allow a display resolution to be selected from a predetermined number of resolutions. Generally, frame buffers are secured in continuous areas
15 in a memory, and therefore, data in the frame buffers must first be collected to form a frame when carrying out a collaborative rendering operation with, for example, a distributed shared memory computer system.

20 SUMMARY OF THE INVENTION

A first aspect of the present invention provides a video output controller having a first storage device configured to store a DMA command list containing DMA commands, a second storage device configured to secure
25 frame buffers, a video output buffer, a DMA controller, and a display controller, wherein the display controller

has a DMA command list processor configured to determine which of the DMA commands contained in the DMA command list must be issued, an initialize signal port configured to receive an initialize signal for starting initialization, a step signal port configured to receive a step signal for starting the issuance of the DMA command, and an external signal processor configured to provide the DMA command list processor with a timing signal for issuing a DMA command according to the initialize signal and step signal.

A second aspect of the present invention provides a video output controller having a video output buffer, a DMA controller, and a display controller, wherein the display controller has a DMA command list processor configured to determine which of the DMA commands contained in the DMA command list must be issued, an initialize signal port configured to receive an initialize signal for starting initialization, a step signal port configured to receive a step signal for starting the issuance of the DMA command, and an external signal processor configured to provide the DMA command list processor with a timing signal for issuing a DMA command according to the initialize signal and step signal.

A third aspect of the present invention provides a video card having a plurality of image processors

configured to draw an image and generate image data to be stored in frame buffers, a first storage device configured to store a DMA command list containing DMA commands, a second storage device configured to secure
5 the frame buffers, a video output buffer, a DMA controller, and a display controller, wherein the display controller has a DMA command list processor configured to determine which of the DMA commands contained in the DMA command list must be issued, an initialize
10 signal port configured to receive an initialize signal for starting initialization, a step signal port configured to receive a step signal for starting the issuance of the DMA command, and an external signal processor configured to provide the DMA command list processor
15 with a timing signal for issuing a DMA command according to the initialize signal and step signal.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram showing a video card
20 according to a first embodiment of the present invention;

Fig. 2 is a block diagram showing a part of the video card of Fig. 1 corresponding to a video output controller;

Fig. 3 is a flowchart showing a basic operation of
25 the video output controller of Fig. 2;

Fig. 4 is a view showing a state of outputting data

from a plurality of frame buffers to a screen under the control of the video output controller of Fig. 2;

Fig. 5 is a block diagram showing a video output controller according to a modification of the first
5 embodiment;

Fig. 6 is a block diagram showing a video output controller according to a second embodiment of the present invention;

Fig. 7 is a flowchart showing a basic operation of
10 the video output controller of Fig. 6;

Fig. 8 is a view showing a state of outputting data from a plurality of frame buffers to a screen under the control of the video output controller of Fig. 6;

Fig. 9 is a block diagram showing the details of a
15 structure of DMA command groups shown in Fig. 6;

Fig. 10A is a view showing a background according to a modification of the second embodiment;

Fig. 10B is a view showing a first window according to the modification of the second embodiment;

20 Fig. 10C is a view showing a second window according to the modification of the second embodiment;

Fig. 10D is a view showing an example of a screen in which the first and second windows are superposed on the background;

25 Fig. 11 is a block diagram showing the details of a structure of DMA command groups according to the

modification of the second embodiment;

Fig. 12 is a block diagram showing a video output controller according to a third embodiment of the present invention;

5 Fig. 13 is a view showing a simple algorithm for vertical-horizontal double enlargement;

Fig. 14 is a view showing a bi-linear algorithm for vertical-horizontal double enlargement;

Fig. 15A is a view showing a simple algorithm for
10 horizontal 1/2 reduction;

Fig. 15B is a view showing a simple algorithm for vertical 1/2 reduction;

Fig. 16 is a view showing a bi-linear algorithm for vertical-horizontal 1/2 reduction; and

15 Fig. 17 is a view showing examples of block flags and DMA commands according to the third embodiment.

DETAILED DESCRIPTION OF EMBODIMENTS

Various embodiments of the present invention will
20 be described with reference to the accompanying drawings. It is to be noted that the same or similar reference numerals are applied to the same or similar parts and elements throughout the drawings, and the description of the same or similar parts and elements
25 will be omitted or simplified.

FIRST EMBODIMENT

Figure 1 shows a video card 1 according to the first embodiment of the present invention. The video card 1 has a plurality of image processors (such as rendering engines) 4a to 4d to draw an image and generate image data to be stored in frame buffers FB1 to FB4, a first storage device 8 to store a DMA command list, a second storage device 9 to secure the frame buffers FB1 to FB4, a display controller 7 to control output to a display unit 3, a video output buffer 5, and a DMA controller 6.

Figure 2 shows a part of the video card 1 shown in Fig. 1 corresponding to a video output controller. The video output controller includes the first storage device 8 to store the DMA command list 16, the second storage device 9 to keep the frame buffers FB1 to FB4, the video output buffer 5, the DMA controller 6, and the display controller 7. The DMA command list 16 includes a plurality of DMA commands DC1 to DC5 each containing a source address (src addr), a destination address (dst addr), and the size (size) of data to be transferred from a frame buffer. The display controller 7 includes a DMA command list processor 10 to determine which of the DMA commands DC1 to DC5 is issued, an initialize signal port 13 to receive an initialize signal SI for starting initialization, a step signal port 14 to receive a step signal Ss for starting the issuance of the DMA

commands DC1 to DC5, and an external signal processor 15 to provide, in accordance with the initialize signal SI and step signal Ss, the DMA command list processor 10 with a timing signal for issuing the DMA commands DC1 to DC5. The DMA command list processor 10 includes a list header 11 to hold a start position of the DMA command list 16 and a list pointer 12 to hold the position of a DMA command to be issued next. The initialize signal SI corresponds to a so-called V-sync signal, and the step signal Ss to a so-called H-sync signal. The display controller 7 receives the initialize signal SI and step signal Ss.

The external signal processor 15 generates a timing signal for the DMA command list processor 10 in response to the initialize signal SI and step signal Ss. More precisely, the external signal processor 15 detects drawing completion timing and provides an initialize timing signal to the DMA command list processor 10. Also, the external signal processor 15 provides a DMA command issuance timing signal in response to an external signal.

The DMA command list processor 10 picks up a proper one of the DMA commands DC1 to DC5 from the DMA command list 16 stored in the first storage device 8 and transfers the DMA command to the DMA controller 6. Namely, the DMA command list processor 10 issues a

DMA command to the DMA controller 6 according to information from the external signal processor 15.

The DMA controller 6 receives the DMA commands DC1 to DC5 directly from the display controller 7. Alternatively, the DMA controller 6 indirectly receives the DMA commands DC1 to DC5 with the use of a pointer. In this case, the DMA controller 6 reads the DMA commands DC1 to DC5 from the first storage device 8 according to the pointer. Based on the read DMA commands, the DMA controller 6 carries out block transfer. Each of the DMA commands DC1 to DC5 stored in the first storage device 8 and activated in a given step may consist of a plurality of DMA commands.

The video output buffer 5 receives image data from the frame buffers FB1 to FB4 and temporarily stores the image data for video output. The video output buffer 5 usually has a capacity to keep data for one line.

The first storage device 8 may be a memory to be directly accessed by the display controller 7 at high speed, or an area secured in the second storage device 9. The elements (DMA commands DC1 to DC5) of the DMA command list 16 have each DMA command data and a pointer for DMA command data to be issued in response to next reception of the step signal Ss. To update the list pointer 12, the list pointer 12 is rewritten with a pointer for DMA command data. The elements of the

DMA command list have an identical size and are aligned in order of activation. Updating the list pointer 12 equates to adding the size of an element of the DMA command list 16 to the list pointer 12.

5 The second storage device 9 is preferably an internal DRAM such as an embedded DRAM.

A basic operation of the video output controller will be explained with reference to Fig. 3.

(a) In step S01, the initialize signal port 13
10 receives the initialize signal SI (V-sync) to start initialization. In step S02, the DMA command list processor 10 is initialized. More precisely, the DMA command list processor 10 copies a start position held in the list header 11 to the list pointer 12. Namely, the
15 DMA command list processor 10 loads a DMA command list start address stored in the list header 11 to the list pointer 12.

(b) In step S03, the step signal port 14 receives the step signal Ss to start issuing the DMA commands
20 DC1 to DC5. In step S04, the display controller 7 transfers the DMA commands DC1 to DC5 to the DMA controller 6. According to the DMA commands DC1 to DC5, the DMA controller 6 transfers image data from the frame buffers FB1 to FB4 of the second storage device 9
25 to the video output buffer 5. Namely, the display controller 7 sends the DMA commands DC1 to DC5 to the

DMA controller 6 to conduct block transfer. In practice, the DMA controller 6 transfers data for one line to the video output buffer 5. In more detail, when the step signal port 14 receives the step signal Ss, the DMA
5 command list processor 10 issues the DMA commands DC1 to DC5 specified by the list pointer 12 and updates the list pointer 12 according to predetermined rules.

(c) Step S05 determines whether or not a frame has completely been drawn. When the frame has been
10 drawn (Yes in step S05), the flow returns to step S01, and the initialize signal port 13 waits for the initialize signal SI (V-sync) to start initialization and draw a new frame. When the frame has not been drawn yet (No in step S05), the step signal port 14 waits for the step
15 signal Ss to start issuing the DMA commands DC1 to DC5.

In this way, the DMA commands DC1 to DC5 to instruct frame data transfer can freely be defined in the first storage device 8 by application. Even if the frame
20 buffers FB1 to FB4 are freely distributed in the second storage device 9 as shown in Fig. 4 according to distributed rendering, image data in the frame buffers FB1 to FB4 can be displayed on an output screen 20 of the display unit 3 of Fig. 1 without collecting the frame
25 buffers FB1 to FB4 to one location.

Since the frame buffers FB1 to FB4 can freely be

distributed in the second storage device 9, a memory space used in the second storage device 9 can be minimized.

5 The video card 1 or the video output controller operates in response to the initialize signal SI and step signal Ss that are externally provided. This results in suppressing distortion of images displayed on the output screen 20.

10 In this way, the first embodiment of the present invention is capable of outputting image data stored in the frame buffers FB1 to FB4 distributed in the second storage device 9 without collecting the frame buffers FB1 to FB4 in one location. This improves the degree of freedom of rendering conducted by application and provides, with a smaller memory space, the same effect as the related art that must collect the frame buffers to one location. According to the related art, the functions of the video output controller such as a CRT control function have been entirely realized by hardware.

15 20 The first embodiment allows some of the functions to be replaced with software and a display resolution to be freely changed to another. The first embodiment can flexibly cope with frame buffers distributed in a memory.

25 An example of the display unit 3 is a CRT display unit, a liquid crystal display unit, or a plasma display unit. The display controller 7 of Fig. 2 is applicable to

a controller (for example, a CRT controller) of these display units.

MODIFICATION OF FIRST EMBODIMENT

5 The initialize signal (V-sync) and step signal (H-synch) are periodical signals. The initialize signal SI is activated at the intervals of a predetermined number of activations of the step signal Ss. A modification of the first embodiment of the present
10 invention employs a counter to count the number of times of reception of the step signal Ss at the external signal processor 15.

Figure 5 shows a video output controller according to the modification of the first embodiment. The video
15 output controller has a first storage device 8 having a DMA command list 16, a second storage device 9 to secure frame buffers FB1 to FB4, a video output buffer 5, a DMA controller 6, and a display controller 7. The DMA command list 16 includes a plurality of DMA
20 commands DC1 to DC5. The display controller 7 includes a DMA command list controller 10 to determine which of the DMA commands DC1 to DC5 must be issued, an initialize signal port 13 to receive an initialize signal SI for starting initialization, a step signal port 14 to
25 receive a step signal Ss for starting the issuance of the DMA commands DC1 to DC5, and an external signal

processor 15 to provide the DMA command list processor 10 with a timing signal for issuing the DMA commands DC1 to DC5 according to the initialize signal SI and step signal Ss. The DMA command list processor 10 includes
5 a list header 11 to keep a start position of the DMA command list 16 and a list pointer 12 to keep the position of a DMA command to be issued next.

The external signal processor 15 has a counter 22 to count the number of times of reception of the step
10 signal Ss and a maximum register 21 to beforehand store the number of steps in which drawing is completed. When the number of times of reception of the step signal Ss becomes equal to the number stored in the maximum register 21, the external signal processor 15 sends an
15 initialize timing signal to the DMA command list processor 10. The number stored in the maximum register 21 corresponds to a maximum number of times of reception of the step signal Ss in each frame.

When the number of times of reception of the step
20 signal Ss for a given frame reaches the maximum number, the external signal processor 15 checks to see when the initialize signal SI is received. When no initialize signal SI is received, the external signal processor 15 sends an interrupt to a higher system to notify of the
25 inconsistency.

In this way, the modification of the first

embodiment counts the number of times of reception of the step signal Ss in every frame, in order to correctly display images.

In addition, the modification provides the same
5 effects as the first embodiment.

SECOND EMBODIMENT

The second embodiment of the present invention vertically divides an output screen 20 of a display unit 3
10 and distributes frame buffers corresponding to the divided parts of the screen in a memory.

Figure 6 shows a video output controller according to the second embodiment. The video output controller includes a first storage device 8 having a DMA command
15 list 16, a second storage device 9 securing frame buffers FB1 to FB4, a video output buffer 5, a DMA controller 6, and a display controller 7. The DMA command list 16 includes a plurality of DMA command groups DC1 to DC5. The display controller 7 includes a DMA command list
20 processor 10 to determine which of the DMA command groups DC1 to DC5 must be issued, an initialize signal port 13 to receive an initialize signal SI for starting initialization, a step signal port 14 to receive a step signal Ss for starting the issuance of the DMA command
25 groups DC1 to DC5, and an external signal processor 15 to provide the DMA command list processor 10 with a

timing signal for issuing the DMA command groups DC1 to DC5 according to the initialize signal SI and step signal Ss.

Each of the DMA command groups DC1 to DC5 to be
5 issued in response to the step signal Ss contains a plurality of DMA commands CMD1 to CMD4. Namely, one time of reception of the step signal Ss starts to issue a plurality of DMA commands CMD1 to CMD4. The details of the structure of the DMA command groups DC1
10 to DC5 will be explained later with reference to Fig. 9.

The DMA command list processor 10 has a list header 11 to keep a start position of the DMA command list 16, a list pointer 12 to keep a position of a DMA command group to be issued next, and a command pointer
15 23 to follow DMA commands CMD1 to CMD4 to be activated by the step signal Ss. Upon receiving the step signal Ss, the list pointer 12 switches a DMA command group to be activated to another among the DMA command groups DC1 to DC5. The command pointer 23
20 switches a DMA command to be activated by the step signal Ss to another among the DMA commands CMD1 to CMD4.

A basic operation of the video output controller of Fig. 6 will be explained with reference to Fig. 7.

25 (a) In step S11, the initialize signal port 13 receives the initialize signal SI for starting

initialization. In step S12, the DMA command list processor 10 is initialized. More precisely, the DMA command list processor 10 copies the start position kept in the list header 11 to the list pointer 12.

5 (b) In step S13, the step signal port 14 receives the step signal Ss for starting the issuance of the DMA command groups DC1 to DC5. In step S14, the display controller 7 transfers the DMA command groups DC1 to DC5 to the DMA controller 6. According to the DMA
10 command groups DC1 to DC5, the DMA controller 6 transfers image data from the frame buffers FB1 to FB4 in the second storage device 9 to the video output buffer 5. Namely, the display controller 7 sends the DMA command groups DC1 to DC5 to the DMA controller 6, to
15 conduct block transfer. In practice, the DMA controller 6 transfers data for one line to the video output buffer 5. In more detail, when the step signal port 14 receives the step signal Ss, the DMA command list processor 10 issues the DMA commands CMD1 to CMD4 contained in
20 one of the DMA command groups DC1 to DC5 specified by the list pointer 12 and updates the list pointer 12 according to predetermined rules. The DMA commands CMD1 to CMD4 issued in response to one time of reception of the step signal Ss are simultaneously or
25 sequentially transferred to the DMA controller 6. According to the second embodiment, transfer order and

activation order of the DMA commands CMD1 to CMD4 are not particularly fixed. The DMA commands CMD1 to CMD4 may be activated and processed in parallel, or may be activated in order of transfer.

5 (c) Step S15 determines whether or not a frame has been drawn. When the frame has been drawn (Yes in step S15), the flow returns to step S11, and the initialize signal port 13 waits for the initialize signal SI (V-synch) for starting the initialization and drawing of a
10 new frame. When the frame has not been drawn yet (No in step S15), the flow returns to step S13, and the step signal port 14 waits for the step signal Ss for starting the issuance of the DMA command groups DC1 to DC5.

 In Fig. 8, the output screen 20 of the display unit 3
15 is vertically divided, and the frame buffers FB1 to FB4 corresponding to the divided parts of the output screen 20 are distributed in the second storage device 9. With this configuration, the second embodiment can transport image data to the output screen 20 without collecting the
20 frame buffers FB1 to FB4 to one location.

 The second embodiment allows the frame buffers FB1 to FB4 to freely be distributed in the second storage device 9. This results in minimizing a memory space to be used in the second storage device 9.

25 The video card 1 or the video output controller according to the second embodiment operates in response

to the initialize signal SI and step signal Ss that are externally provided. This results in suppressing distortion of images displayed on the output screen 20.

In this way, the second embodiment of the present invention is capable of outputting image data stored in the frame buffers FB1 to FB4 distributed in the second storage device 9 without collecting the frame buffers FB1 to FB4 to one location. This improves the degree of freedom of rendering conducted by application and provides, with a smaller memory space, the same effect as the related art that must collect the frame buffers to one location. According to the related arts, the functions of the video output controller have been entirely realized by hardware. The second embodiment allows some of the functions to be replaced with software and a display resolution to be freely changed to another. The second embodiment can flexibly cope with frame buffers distributed in a memory.

An example of the display unit 3 is a CRT display unit, a liquid crystal display unit, or a plasma display unit. The display controller 7 of Fig. 6 is applicable to a controller (for example, a CRT controller) of these display units.

As shown in Fig. 9, the DMA command list 16 has a hierarchical structure consisting of a plurality of levels. Namely, the DMA command list 16 includes a plurality of

DMA command groups DC1 to DC5. Each of the DMA command groups DC1 to DC5 includes a plurality of DMA commands CMD1 to CMD4 and a node 25. The list pointer 12 indicates a first level node, for examples, the node 25 of the DMA command group DC3. When the DMA command groups DC1 to DC5 are issued, the DMA commands CMD1 to CMD4 that follow the node 25 of the DMA command group DC4 whose level is lower than the level of the DMA command group DC3 indicated with the list pointer 12 are activated. The list header 11 holds a start position, i.e., the node 25 of the DMA command group DC1 in the DMA command list 16. The list pointer 12 keeps the position of the node 25 of the DMA command group DC3 to be issued next. The command pointer 23 follows DMA commands CMD1 to CMD4 to be activated in response to the step signal Ss.

MODIFICATION OF SECOND EMBODIMENT

A method of displaying window images generated through a plurality of processes on a single screen according to a modification of the second embodiment will be explained. The modification superposes first and second windows 31 and 32 of Figs. 10B and 10C on a background of Fig. 10A and displays them in a single frame as shown in Fig. 10D.

According to the second embodiment of Fig. 9, the

DMA command groups DC1 to DC5 activated in response to the step signal Ss have an identical number of DMA commands CMD1 to CMD4. On the other hand, the modification of the second embodiment provides DMA
5 command groups with different numbers of DMA commands depending on windows to be superposed. Namely, a part (line) 33 where the first and second windows 31 and 32 are displayed over the background 30 needs additional DMA commands to display the first and
10 second windows 31 and 32.

In Fig. 11, DMA command groups DC1, DC4, and DC5 need to display only the background 30, and therefore, each of them contains a node 25 and DMA commands CMD1 and CMD2. On the other hand, DMA
15 command groups DC2 and DC3 must display the first and second windows 31 and 32 over the background 30, and therefore, each of them contains a node 25 and DMA commands CMD1 to CMD4.

Unlike the second embodiment of Fig. 9, the
20 modification of Fig. 11 must maintain the order in issuing the DMA commands CMD1 to CMD4. Namely, the background 30 must be transferred first and then the first and second windows 31 and 32 to be written over the background 30. Accordingly, a command pointer 23
25 must follow the DMA commands CMD1 to CMD4 in this order.

THIRD EMBODIMENT

The third embodiment according to the present invention enlarges and reduces images.

5 Figure 12 shows a video output controller according to the third embodiment. This video output controller includes a first storage device 8 having a DMA command list 16, a second storage device 9 to secure frame buffers FB1 to FB4, a video output buffer 5, a DMA
10 controller 6, a display controller 7, and filtering hardware 40 to conduct filtering for image enlargement and reduction. The filtering hardware 40 includes an intermediate data memory 41 and an interpolater 42. The DMA command list 16 contains a plurality of DMA
15 commands DC1 to DC5. The display controller 7 has a DMA command list processor 10, an initialize signal port 13, a step signal port 14, and an external signal processor 15. The DMA command list processor 10 has a list header 11 and a list pointer 12.

20 Compared with the video output controller of Fig. 2, the video output controller according to the third embodiment additionally has the filtering hardware 40.

 With reference to Figs. 13 to 16, techniques of enlarging and reducing images with the use of the video
25 output controller of Fig. 12 will be explained.

 Figure 13 shows a simple algorithm to double an

image in vertical and horizontal directions. An upper part of Fig. 13 shows a line of pixels before double enlargement, a middle part of Fig. 13 shows a line of pixels after double enlargement in a horizontal direction, and a lower part of Fig. 13 shows lines of pixels after double enlargement in a vertical direction. Although this example relates to double enlargement, the same technique is applicable to enlargement with any integer multiple. In Fig. 13, a line of pixels $C_{1,1}$, $C_{1,2}$, $C_{1,3}$, ... is transferred to the intermediate data memory 41 of the filtering hardware 40, to conduct a horizontal enlarging process on the line of pixels. More precisely, as shown in the upper and middle parts of Fig. 13, an intermediate pixel $C_{1,j+0.5}$ is produced between adjacent ones of the pixels $C_{1,1}$, $C_{1,2}$, $C_{1,3}$, ... in the line. The color of the left pixel is copied to generate the color of the intermediate pixel $C_{1,j+0.5}$. After the completion of the horizontal enlarging process, the line $C_{1,j}$ is transferred twice to the video output buffer 5 to conduct a vertical enlarging process that forms lines $C_{1,j}$ and $C_{2,j}$ as shown in the lower part of Fig. 13.

Figure 14 shows a bi-linear algorithm to double an image in vertical and horizontal directions. Compared with the technique of Fig. 13, an improvement in the quality of images is expected according to the technique of Fig. 14. An upper part of Fig. 14 shows lines of pixels

before double enlargement, a middle part of Fig. 14
 shows lines of pixels after horizontal double
 enlargement, and a lower part of Fig. 14 shows lines of
 pixels after vertical double enlargement. Although this
 5 example relates to double enlargement, the same
 technique is applicable to enlargement with any multiple.
 In Fig. 14, a first line of pixels $C_{1,1}$, $C_{1,2}$, $C_{1,3}$, ... is
 transferred to the intermediate data memory 41 to
 conduct a horizontal enlarging process on the first line
 10 according to an expression (1) mentioned below. More
 precisely, as shown in the upper and middle parts of Fig.
 14, an intermediate pixel $C_{1,j+0.5}$ is generated between
 adjacent ones of the pixels $C_{1,1}$, $C_{1,2}$, $C_{1,3}$, ... in the first
 line. At this time, the value of the intermediate pixel
 15 $C_{1,j+0.5}$ is obtained from the adjacent pixels according to
 linear interpolation. In the case of double enlargement,
 an average of the adjacent two pixels $C_{1,j}$ and $C_{1,j+1}$ is
 simply calculated. Next, pixels $C_{2,1}$, $C_{2,2}$, $C_{2,3}$, ... of a
 second line are transferred to the intermediate data
 20 memory 41, and the horizontal enlarging process is
 conducted on the second line according to an expression
 (2).

$$C_{i,j+0.5} = (C_{i,j} + C_{i,j+1}) / 2 \quad \dots(1)$$

25

$$C_{i+1,j+0.5} = (C_{i+1,j} + C_{i+1,j+1}) / 2 \quad \dots(2)$$

Thereafter, vertical enlargement is conducted between the first-line pixels $C_{1,1}$, $C_{1,2}$, $C_{1,3}$, ... and the second-line pixels $C_{2,1}$, $C_{2,2}$, $C_{2,3}$, More precisely, as
5 shown in the lower part of Fig. 14, linear interpolation is carried out between vertically corresponding pixels of the first and second lines to generate an intermediate pixel $C_{i+0.5,j}$. Next, the first line and the intermediate line corresponding to a "1.5"th line are transferred to
10 the video output buffer 5. Then, a third line of an original image is transferred to the intermediate data memory 41 to similarly conduct the horizontal enlargement and vertical enlargement by generating intermediate pixels. Repeating these processes
15 completes the vertical-horizontal double enlargement.

Figure 15A shows a simple algorithm to reduce 1/2 times an image in vertical and horizontal directions. An upper part of Fig. 15A shows a line of pixels before horizontal 1/2 reduction and a lower part of Fig. 15A
20 shows a line of pixels after the horizontal 1/2 reduction. Although this example relates to 1/2 reduction, the same technique is applicable to reducing "1/integer" times an image. In Fig. 15A, a line of pixels $C_{1,1}$, $C_{1,2}$, $C_{1,3}$, ... is transferred to the intermediate data memory 41 and
25 every second pixels $C_{1,2}$, $C_{1,4}$, ... are removed to form a line of 1/2 size, to complete the horizontal 1/2 reduction.

Figure 15B shows a simple algorithm to reduce $1/2$ times an image in vertical and horizontal directions. A left part of Fig. 15B shows lines of pixels before vertical-horizontal $1/2$ reduction, a central part of Fig. 15B shows lines of pixels after horizontal $1/2$ reduction, and a right part of Fig. 15B shows lines of pixels after vertical $1/2$ reduction. Although this example relates to $1/2$ reduction, the same technique is applicable to reducing "1/integer" times an image. In Fig. 15B, a first line is transferred to the intermediate data memory 41, and every second pixels are removed to form a line of $1/2$ size. This operation is repeated to reduce $1/2$ times a given image in the horizontal direction. The pixels after the horizontal reduction are transferred to the video output buffer 5. This horizontal reduction process is conducted on every second lines, and resultant data is transferred to the video output buffer 5 to complete the vertical reduction process.

Figure 16 shows a bi-linear interpolation algorithm to reduce $1/2$ times an image in vertical and horizontal directions. An upper part of Fig. 16 shows lines of pixels before vertical-horizontal $1/2$ reduction and a lower part of Fig. 16 shows a line of pixels after the vertical-horizontal $1/2$ reduction. At first, pixels in two lines are transferred to the intermediate data memory 41 to generate new pixels each derived from an

average of four pixels, i.e., vertical and horizontal 2 x 2 pixels. This completes the vertical-horizontal 1/2 reduction process. The newly generated line is transferred to the video output buffer 5 to display a
5 reduced image.

The video output controller shown in Fig. 12 requires a transfer synchronization mechanism to first transfer data to the filtering hardware 40. Thereafter, processes in the filtering hardware 40 are started. The
10 processed data is transferred from the filtering hardware 40 to the video output buffer 5. These operations must be carried out in this order. Accordingly, the display controller 7 carries out not only DMA transfer but also activation of the filtering
15 hardware 40. Namely, the DMA command list 16 allows two descriptions, one for the data transfer and another for the hardware activation.

As shown in Fig. 17, the DMA command list 16 is additionally provided with block flags 43. The display
20 controller 7 detects the end of a DMA command and executes the next DMA command. Since the display controller 7 controls transfer order, there is provided a mechanism to notify the display controller 7 of the end of processes carried out by the filtering hardware 40. In
25 addition, the display controller 7 is provided with a mechanism to stop data transfer until the reception of

the process end notification from the filtering hardware 40. According to the third embodiment, each element contained in the hierarchical DMA command list 16 has the block flag 43. When the block flag 43 is "1," the display controller 7 stops executing the corresponding DMA command until the reception of the process end notification from the filtering hardware 40. This arrangement is capable of conducting data transfer to the filtering hardware 40, activation of the filtering hardware 40, and transfer of processed data in predetermined order.

The third embodiment of the present invention is capable of outputting image data stored in the frame buffers FB1 to FB4 distributed in the second storage device 9 without collecting the frame buffers FB1 to FB4 to one location. This improves the degree of freedom of rendering conducted by application and provides, with a smaller memory space, the same effect as the related art that must collect the frame buffers to one location. According to the related art, the functions of the video output controller have entirely been realized by hardware. The third embodiment allows some of the functions to be replaced with software and a display resolution to be freely changed to another. The third embodiment can flexibly cope with frame buffers distributed in a memory.

As mentioned above, the present invention has been described through the first to third embodiments and modifications thereof, however, the descriptions and drawings that constitute a portion of this disclosure should not be perceived as those limiting the present invention. Various alternative embodiments and operational techniques will become clear to persons skilled in the art from this disclosure.

Therefore, the present invention in its broader aspects is not limited to the specific details and representative embodiments shown and described herein. Accordingly, various modifications may be made without departing from the spirit or scope of the general inventive concept as defined by the appended claims and their equivalents.